Using Simulation Based Design for Enhanced Performance and Fault Tolerance

Kenneth G. Blemel
Management Sciences, Inc.
6022 Constitution Ave. NE
Albuquerque, NM 87110-5941
Phone: (505) 255-8611 • Fax: (505) 268-6696
Email: ken_blemel@mgtsciences.com

ABSTRACT: This paper describes how Defense Advanced Research Projects Agency (DARPA) rapid prototyping projects are paying off in new ways to enhance performance and fault tolerance with simulation based design (SBD). The projects have resulted in major technology advances that provide knowledge based methods for dynamic health and status management through strategic planning as a part of simulation based design. SBD is used to address the most fundamental challenges of fault tolerant strategies, principles and practices: how to use the power of software of embedded processors to enhance fault tolerance without restricting performance. This paper outlines a simple, elegant, and robust methodology that can use SBD to enhance performance and extend the life of multi-processor systems, and networks of systems with embedded processors.

1.0 Using Smart Reactive Supervisors for Health Management

Achieving robust performance and fault tolerance in processors is a lot like human life extension and life support. Monitoring, diagnosis, prognosis, treatment and an adaptive strategy for optimum life extension are all important. The watching of events in real-time systems is called health status monitoring. Monitoring is usually simple, deterministic, and static. Monitoring is performed by patient monitors and flight recorders that log the events and effects, but do little to diagnose, prognose, or react to optimize outcomes. Reactive health management includes pro-active measures to conserve and allocate critical resources to extend life and achieve optimum quality of service.

2.0 Implementing Reactive Health Management with Embedded Diagnostics

Health status management is a continuous process, and the effectiveness of treatment can be verified at each task interval. Smart Reactive Supervision (SRS) is a new software technique to optimize the health and performance of real time embedded systems. SRS is a revolutionary change from traditional reliability and testability techniques which include hardening of electronics, reduced electronic stress, and hardware redundancy. SRS provides software implemented fault tolerance (SIFT) and reactive planning as an integral part of the concurrent design (co-design) process. SRS implements the tactics of a robust predefined strategy for flexibility in normal conditions, and extended reliability and performance in degraded or hazardous situations. SRS is used to dynamically refine or modify the current strategy considering current requirements, threats and remaining resources.

SRS is founded on four leading edge technologies:

ESD	- embedded self diagnosis added automatically during coding,
ESP	- embedded self prognosis using knowledge based algorithms,
DRM	- run time resource management to implement strategies and
GTS	- generational (dynamic) task scheduling using genetic algorithms.

Each technology takes advantage of available processing functionality. Each technology can extend the life and functionality of a processor or a set of processors. Combining all four technologies results in synergism to effect smart reactive management through reactive software implemented fault tolerance (SIFT). The breakthrough comes from the ability to autocode the software of embedded processors as the result of simulation based design. SBD also makes it possible for diagnostics to be automatically embedded during software autocoding. ESP uses the diagnostics to isolate detected faults and failures and make immediate prognosis. Based on the ESP, SRS implements triage actions to minimize the damage effects, and restore normalcy using dynamic resource allocation to achieve optimum task schedules determined by GTS.

The real time diagnostic information is used with planning information in the knowledge database and data from the mission requirements database to create SRS. SRS are software agents designed to assure that all tasks are performed according to their relative criticality. Diagnostics embedded in the process algorithms and controls provide feedback to the reactive supervisor about the health and status of resources and processes in light of threats, degradation and opportunities. The SRS uses the feedback to evaluate trends and establish a prognosis of the effect of current operations and anticipated future conditions. The task scheduler includes tasks to mitigate or otherwise provide managed health care. SRS uses diagnostic data generators and diagnosis algorithms embedded in programming using new automatic coding (autocoding) technology. On fault or failure events, the reactive supervisor directs the task scheduler to include tasks for using advanced instrument controllers to manage detrimental effects of failure, e.g. rising thermal conditions leading to intolerable conditions.

3.0 Simulation Based Design for Health Status Management

The Defense Advanced Research Projects Agency (DARPA) has begun to focus on solutions to the fault tolerant systems by using simulation based design (SBD) to implement smart reactive supervisors (SRS) technology. SBD begins in requirements definition as a part of the system engineering process, and continues during architecture definition, trade off studies and detailed design.

3.1 Step 1 Requirements Capture

The process begins with capture of requirements in system engineering. For the DARPA project we used Ascent Logic Corporation’s RDD-100. The requirements are parsed and connected into the requirements schema as hierarchy (dependency) charts that show the derivation of each requirement. The requirements usually include goals for diagnosability, detectability, testability, restoreability, reliability, maintainability, restorability, availability, and safety.

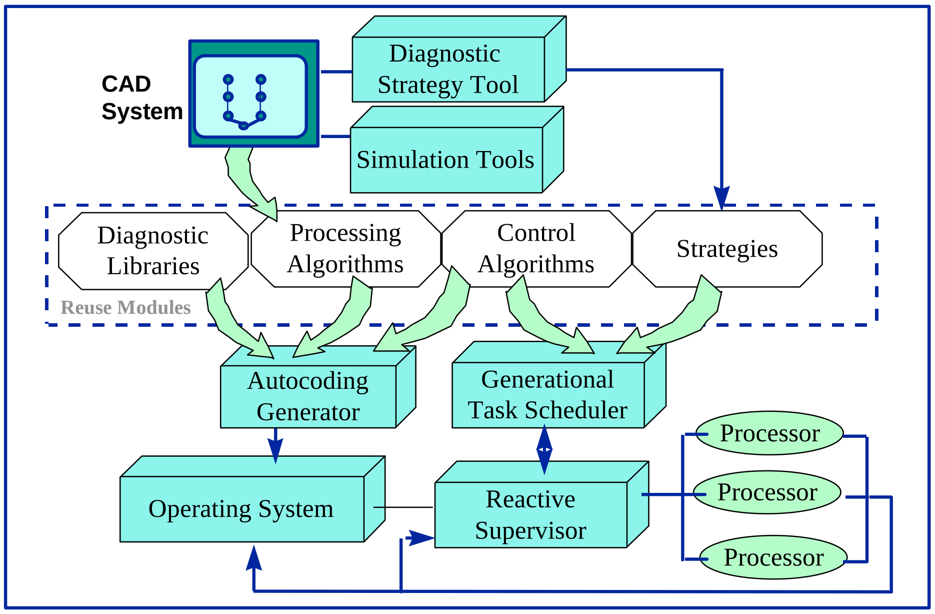


Figure 1. Simulation Based Design Toolset

3.2 Step 2 Behavioral Simulation of Requirements

The simulation of idealized requirements is the first step in SBD, because flawed requirements are the major cause of flawed designs, and requirements have been known to change. The requirements are validated using a simulation technique called operational modes and effects criticality analysis (OMECA). The following figure shows the flow from requirements to implemented SRS.

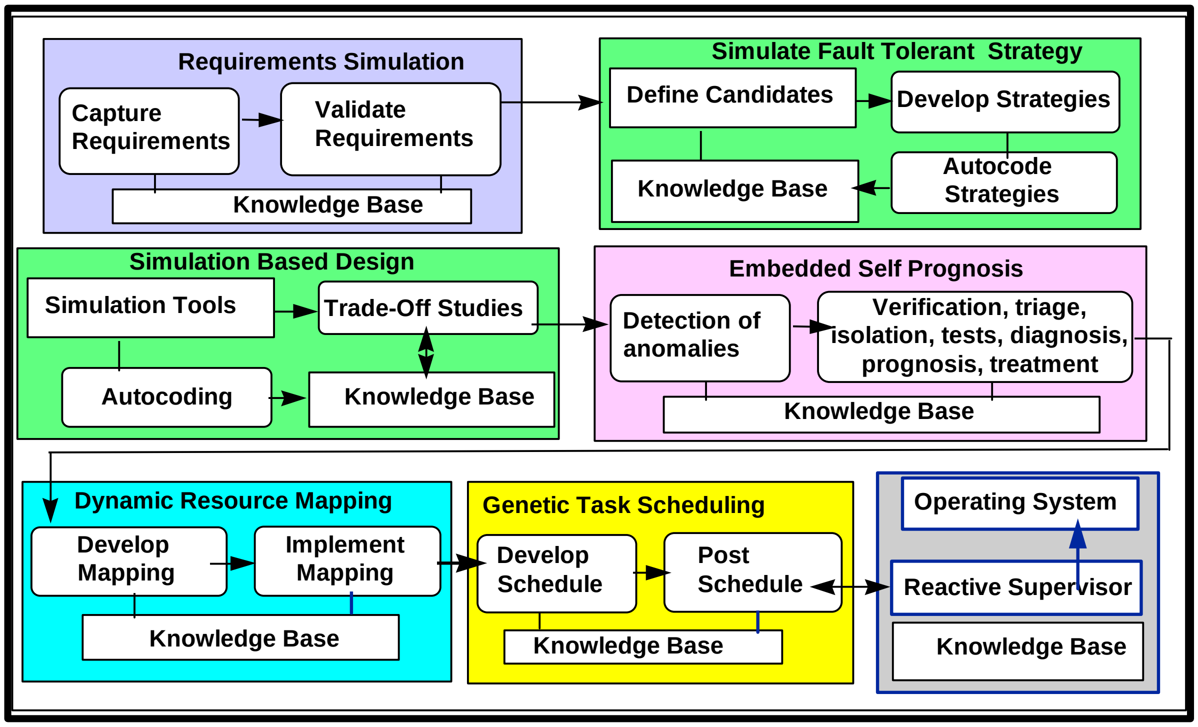
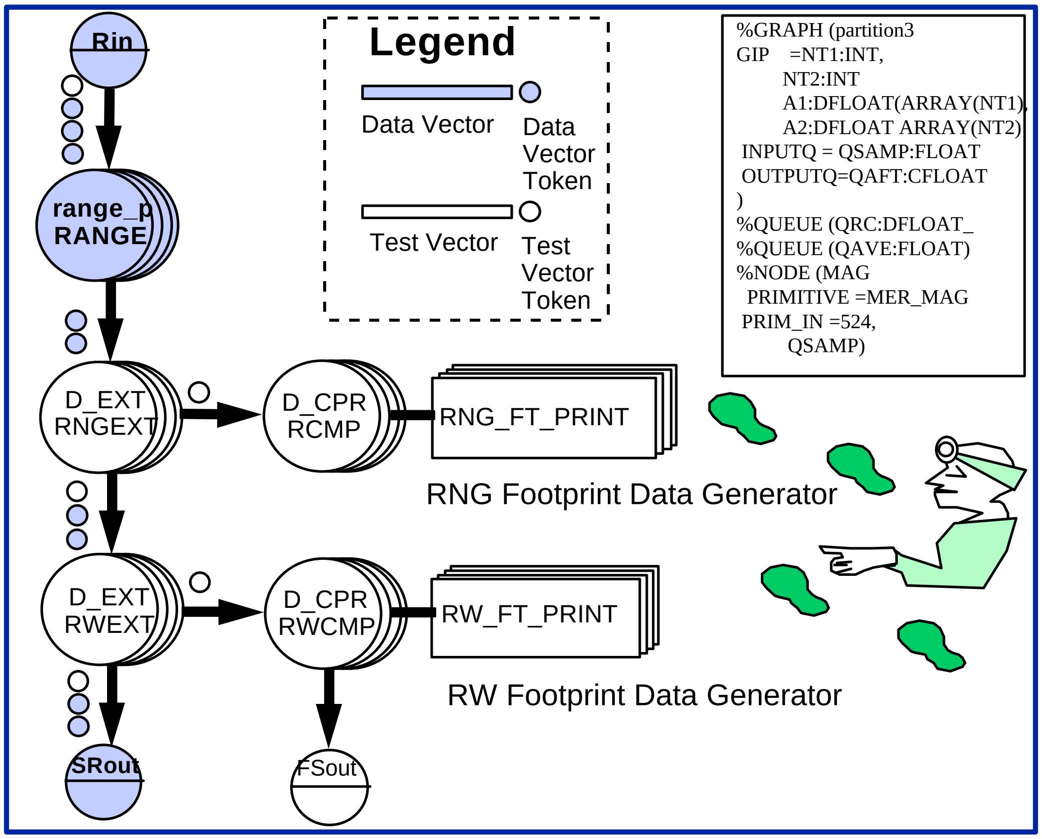


Figure 2. Flow of Design Processes for Enabling Smart Reactive Supervision

The simulation is performed on the requirements schema captured in popular requirements tracking tools. The OMECA is used to understand the ramifications and relative importance of meeting requirements, and the negative aspects of failing to meet requirements. Once the requirements are certified, they are transferred to Computer Aided Design (CAD) functional diagrams. Architecture candidates are added to the functional diagrams by using VHDL or other high level languages. The DARPA Rapid Acquisition of Application Specific Signal Processors (RASSP) project has funded the Computer Executable Electronic Native System Specification (CEENSS) project to reduce the ambiguities and flaws of specifications.

3.3 Step 3 SBD of Diagnostic Strategies

The basic concept of SRS is automatic coding (autocoding) of diagnostics that satisfy the diagnostic strategy as part of the autocoding of the algorithms and control program. The diagnostics are checked at appropriate points defined in the diagnostic strategy. The diagnostics either confirm operation or detect anomalies. The diagnostics can be checking for proper calculations, completion of tasks by processors, detection of threats, attempts to countermeasure, or a variety of other possible conditions. If neural sensor feedback is a part of the overall strategy, these computations are also used in diagnosis. SBD validates and verifies compliance of the diagnostic strategy. The validated process is called "embedded self diagnosis" (ESD). The diagnostic information is stored in the run-time knowledge database. The diagnostic data generators are added and simulated with the Management and Communication Control, Inc. (MCCI) autocoding toolset.



```
%GRAPH (partition3
GIP   =NT1:INT,
        NT2:INT
        A1:DFLOAT(ARRAY(NT1),
        A2:DFLOAT ARRAY(NT2)
 INPUTQ = QSAMP:FLOAT
 OUTPUTQ=QAFT:CFLOAT
)
%QUEUE (QRC:DFLOAT_
%QUEUE (QAVE:FLOAT)
%NODE (MAG
 PRIMITIVE =MER_MAG
 PRIM_IN =524,
         QSAMP)
```

Figure 3. *In-Situ* Diagnostic Data Generators

3.4 Step 4 SBD of Architecture Candidates (Trade-off Studies)

Once the requirements are validated, the process moves to definition of candidate architectures. As in politics, all candidates are somehow imperfect. There are designs with proven technology that will become obsolete, designs with new technology that are unproved, and designs with leading edge technology that have higher risks and potentials for bigger payoffs. Each candidate architecture has "good news and bad news". System engineers and the designers use behavioral (functional) design simulation to help determine which candidate best meets requirements for performance, cost, upgradeability, supportability, reliability, availability, safety, and other independent variables. The results of simulations provide parameters for trade-off studies. For example, the simulations estimate additional overhead for the coding imposed by the diagnostic and health management strategy.

3.5 Step 5 SBD of Prognostic Strategies

The ESD knowledge database and the mission requirements database are the basis for ESP. The mission requirements database includes critical processes for each mission sequence. The diagnostic database contains current information on faults, failures, processes and processors. The ESP routines consider the potentials for mission success in the current situation of resources and threats.

3.6 Step 6 SBD of Task Management Strategies

Tasks are the granular activities that take place during processes. Human managers manipulate tasks of their subordinates at random to achieve changing priorities. The art of task scheduling has been an inherent part of life since the beginning of time. The science of task scheduling is a basic part operations research. Automated task scheduling (ATS) is used to schedule examinations that compete for time and classrooms. Annealed ATS (AATS) stems from the requirement to determine schedules within a specified time. Genetic ATS solves the scheduling problem with distributed computing resources. The next figure shows managing tasks as a function of time.

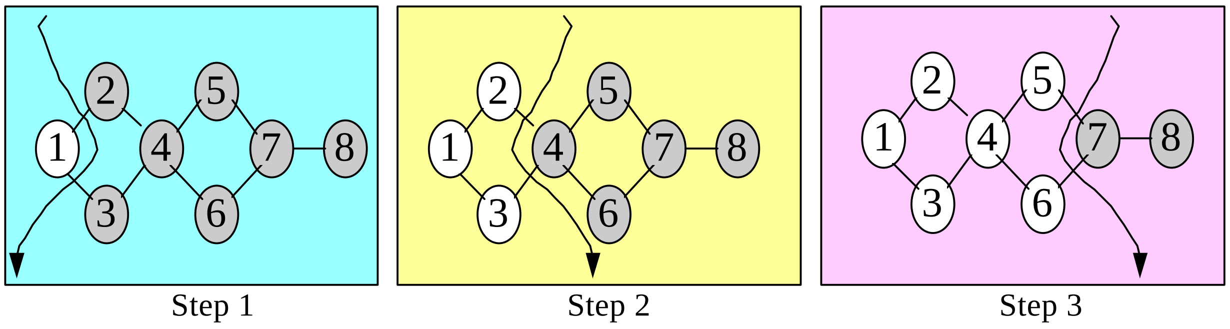


Figure 4. Generational Task Scheduling

GTS is used with ESP and DRM to dynamically optimize current and future functionality according to prevailing strategies and resources. The schedules deal with "first aid" triage plans that will be used in the event of critical failures, and task priority mappings for generating task schedules in real time to optimize performance in the specific mission. SBD is used to validate and verify compliance with the health management strategy. The validated process is called "generational task schedules" (GTS). The GTS strategy and triage actions are stored in the run-time knowledge database.

3.7 Step 7 SBD of Dynamic Resource Management (DRM) Strategies

The ESP knowledge database and the mission requirements database are the basis for reactive resource management. SBD is used to develop the "first aid" triage plans that will be used in the event of critical failures, and task priority mappings for generational task scheduling (GTS) to optimize performance in the specific mission. SBD is used to validate and verify compliance with the health management strategy. The validated process is called "dynamic resource management" (DRM). The DRM strategy and triage actions are stored in the run-time knowledge database.

4.0 Applications

SRS is easily extended to neural sensor networks by using advanced instrument controllers (AIC) and mechanical instrument micro-systems (MIMS). AIC are postage stamp sized hybrid computers that provide health status management and messaging. MIMS are tiny, functional miniature relays, motors, and other mechanic devices mounted on postage size substrates. AIC and MIMS provides real time analysis and control of tasks and processes needed to perform reactive situation management beyond simply using software to monitor, react and control.

4.1 Applications with Advanced Instrumentation Controllers (AIC)

The smallest data elements come from advanced instrument controllers (AIC). An AIC uses multi-chip module (MCM) technology to combine a central processor, A/D and D/A converters, memory, and power conversion into a package no bigger than a commemorative postage stamp. The AIC are a multi-featured, general purpose microprocessor based on 8031 technology. The AIC opens new applications for "smart reactive supervision" with diagnosis, prognosis and generational scheduling that can be extended to distributed "smart" servo mechanisms, sensors and actuators.

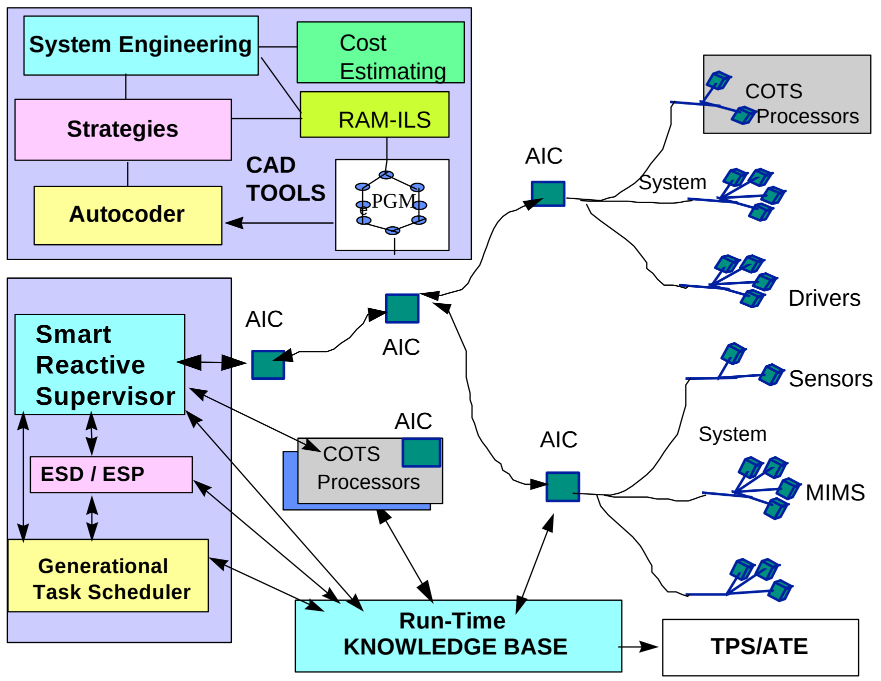


Figure 5. Reactive Supervision with Distributed AIC

AIC are very versatile devices with all the power of a microprocessor and sets of analog to digital (A/D) and digital to analog (D/A) signal converters. With both forms of conversion, the AIC can also perform Analog to Analog conversion, frequency modulation, and digital conversion. AIC are the equivalent of personal hybrid computers in the space of a postage stamp.

AIC can be used with any form of software implemented architecture. Each AIC provides the means by which pro-active and reactive measures can be performed by distributed “smart” servo mechanisms, sensors and actuators. AIC can be used to control devices like micro instruments, switches, relays, and motors. AIC can be used to validate or verify actions performed by the devices under their control. AIC can also be connected to sensors of vital signs such as temperature, velocity, acceleration, pressure, and humidity. AIC can also be used to control redundancy. AIC are switched on or off by control logic, remote signals or power sources.

4.2 SRS Applications with MIMS

MIMS are one or more miniature electro-mechanical devices, and their control circuits are mounted on microchips. MIMS motors, relays, and switches are used to perform actions formally requiring much larger devices. MIMS provide the SRS with “fingertips” to make adjustments and take reactive steps to manage health. For example, MIMS relays can isolate electrical problems, and MIMS motors can position laser mirrors.

4.3 SRS Applications using Dynamic Autocoding of AIC

AIC have important ability to use new coding that can be transferred to the AIC as a task or instruction from the reactive supervisor. Autocoding generates coding of the AIC processors to overcome sticky, stuck actuators or other processes. Coding or commands could be received from external sources such as JAVA BEANS.

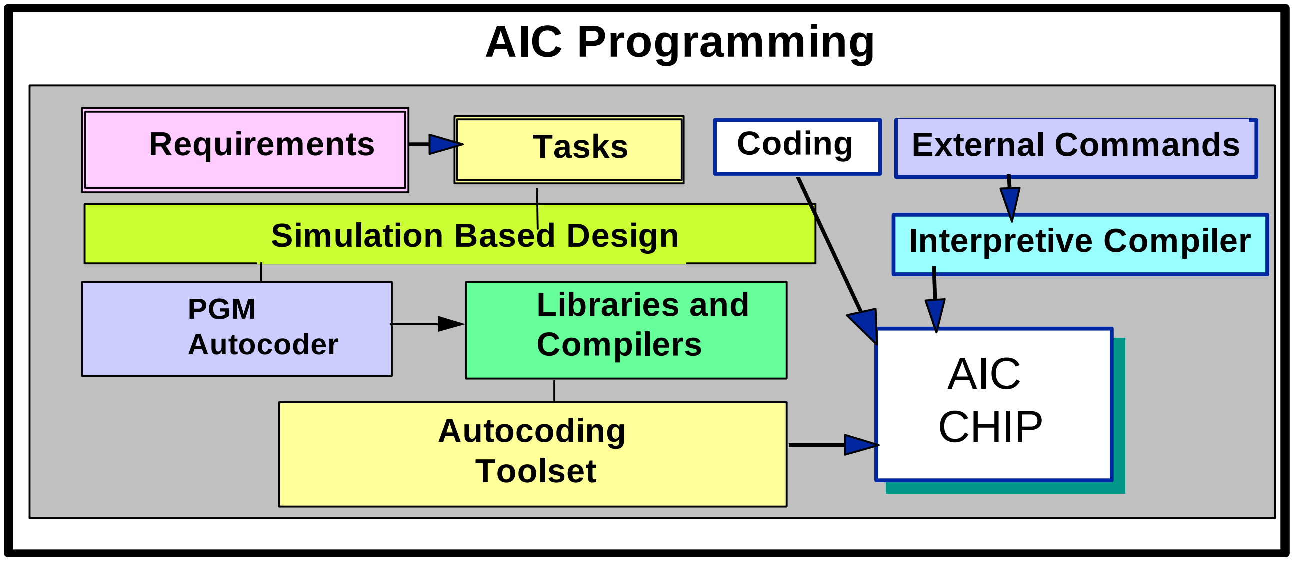


Figure 6. AIC Programming Processes

5.0 Conclusions

These are a few of the many facets of using ESP with AIC for health and status management and maintenance. ESP is definitely feasible in multi-processor applications with AIC. We are confident that the first implementations in laboratory situations will illuminate other benefits of ESP and AIC in areas that extend into other areas than fault tolerance. Opportunistic use of ESP with AIC may yield benefits that are of equal or greater significance when highly fault tolerant, agile use of ESP and AIC is applied in Joint Strike Fighter, remotely piloted vehicles, and hundreds of other applications.

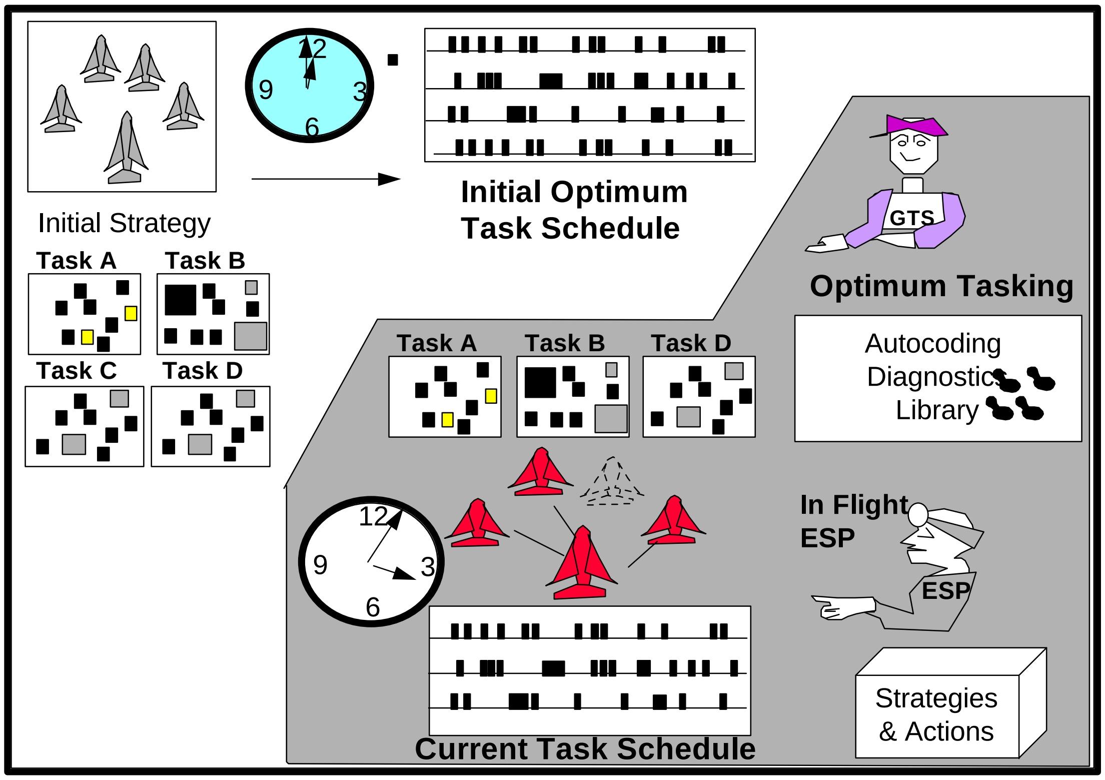


Figure 7. An application of ESP in management of Remotely Piloted Vehicles

Acknowledgments:
ESP was developed under a DARPA Small Business Innovative Research (SBIR) project. ESP is an extension of the co-design methodology developed under the DARPA RASSP project[1]. Technical assistance is being provided by members of the RASSP project team. The SBIR team consists of Management Sciences, Inc. (MSI), Lockheed Martin Corporation Advanced Technology Laboratory (ATL), Management Communications and Control, Inc. (MCCI), Honeywell Corporation, and Utah State University (USU). The DARPA contract monitor is Dr. Jose Munoz (703) 696-4468. In Phase II, SRS with ESP will be applied to DoD digital signal processor systems with benchmarks performed by an independent agency.

[1] Proceedings of the First Annual RASSP conference, August 1994